<바닐라JS로 크롬 앱 만들기>

출처: [Nomad Coders](https://www.youtube.com/channel/UCUpJs89fSBXNolQGOYKn0YQ)의 유튜브 강의 중, [JS Basics](https://www.youtube.com/playlist?list=PL7jH19IHhOLM8YwJMTa3UkXZN-LldYnyK)

#0-1.,#0-2#1-1

자바스크립트는 웹에 쓰이는 하나뿐인 프로그래밍 언어이다.

자바스크립트는 웹사이트를 interactive 하게 만들고 싶을 때 사용하는 것

전 세계의 컴퓨터에서 Js는 돌아간다.

왜?-> 모든 컴퓨터에는 브라우저가 있다.-> 브라우저는 JS에서 돌아간다.-> 모든 컴퓨터에 JS가 깔려 있다. ->모든 컴퓨터가 JS를 이해한다.

#1-3

Js의 version

ES5=ECMAScript5, ES6=ECMAScript6

ECMAScript는 Specification의 명칭이다.

Specification: Js는 중앙집권화 되어 있어서 누군가 업데이트를 하면 모든 브라우저에서 작동을 하게 된다. JS는 specification인데 체계 매뉴얼 같은 것 간단하게 안내책자 같은 것이라고 할 수 있음 (많은 사람들에게 검증 받은 설명문)

브라우저는 specification을 받아서 자기들 방식으로 실행하는 것이다. 브라우저들은 같은 결과를 이루려고 노력하는데 각자의 방식으로 하는 것이다.

ES5,ES6같은 경우 그냥 specification의 버전을 나타내 주는 것이다.

EX> 구글 크롬이 ES6을 지원한다-> 구글 크롬 개발자들이 새로운 specification을 시행했다고 볼 수 있는 것이다.

바닐라JS: JS에서 라이브러리가 없는 것을 의미 한다,

#1-4

JS를 쓰는 많은 웹사이트를 전부 업데이트 해줄 수 없기 때문-> 업데이트 할 경우 수많은 에러 당첨!

JS는 브라우저를 통해 우리에게 제공된 JS라고 할 수 있다.

EX) 매우 현대적이지만 지지 받지 못한 JS를 만듦-> 이것을 compiler안에서 실행 시킴

코드를 상자 안에 넣었다고 상상하자-> 그 박스가 컴파일 된 JS를 전부 모든 브라우저에 뱉어낸다-

현대적인 JS(라이브러리 충만한 , 프레임 워크들을 의미하는 것 같음)을 만들었지만 받게 되는 것은 컴파일된 못생긴 JS인 것이다.

바닐라JS는 모든 웹 언어의 중심이다.

#1-5

JS파일은 항상 Body아래에 있어야 한다.

JS파일 추가가 간단한 이유는 모든 웹 브라우저에는 JS가 설치되어 있기 때문

* A라는 언어로 제작된 프로그램을 다운 받을 시, A라는 언어도 다운 받아야 한다. 컴퓨터가 A라는 언어를 이해 하도록 하기 위해서 하지만, JS에서는 그럴 필요가 없다. 모든 웹 브라우저에는 JS가 설치되어 있기 때문
* 브라우저는 우리가 쓴 명령어를 알아들어서 실행 했을 뿐이다.

#1-6

Variable

JS는 훈육하지 않는 아빠와 같다. => 문법에 엄격하지 않다는 뜻 , 잘못된 문법도 용인한다.

변수를 만들고 싶을 때, 변수는 이렇게 작동한다.

1. 변수를 만든다.
2. 초기화 한다.
3. 사용한다

생성과 초기화를 같이 할 수 도 있다.

Let: 생성시에 쓴다.

#1-7

Const: 상수, 바꿀 수 없다.

Let: 변수가 바뀌는 것을 허용

Var: variable, let과 마찬가지로 값을 바꿀 수 있음

#1-8

저장한다= 대입한다.

Const what=” ”//string

Const what=true//boolen

Const what=899//number

Const what= 44.1//float

#1-9

Array: 데이터를 저장하는 곳 리스트와 같다

Const daysOfWeek=[]//camel case방식: 소문자로 시작해서 스페이스가 필요한 글자를 대문자로

[]기호는 Array의 의미

Array의 값을 가져오려면 첫 번째 요소는 0부터 시작한다.

#1-10

Object

Const person = {

Name: ”Nico”,

Age: 33,

Gender: ”Female”,

isHandsome: true;

};

Object는 Array와 같이 동작하지 않는다.

실제 객체를 만드는 것, object는 실제 값에 라벨을 부여해서 구별해주는 역할이다.

여기서, name, age, gender, isHansome은 변수이다.

Person객체 안에 있는 변수 값은 객체가 const여도 바꾸는 것이 가능하지만, 객체 자체를 바꿀 수는 없다.

데이터를 정렬하는 방법은 Array와 object가 있다.

DB에서 가져온 리스트데이터라면, Array를 선택하고, object를 Array안에 넣을 수 있다.

예를 들어, ,많은 사람들의 정보가 필요한 경우 person객체의 Array를 만들 수 있다.

또한 object안에 ARRAY를 넣는 것도 가능하다.

JS에서 오류가 나도, Html과 CSS는 멀쩡하다.

#2-1

console이라는 object가 있고, log라는 키가 있다. 이 경우에는 log는 function인 것이다.

Console.log():console이라는 오브젝트 안에 들어있는 함수 log를 부르는 것

브라우저가 가지고 있는 자바스크립트는 object에 넣을 수 있다.=>>???  
내장함수

함수: 어떤 것을 수행하려는 한 부분으로, 내가 원하는 만큼 쓸 수 있다. 내가 원하는 만큼 쓸 수 있는 코드

인자(argument): 변수, sayhello()가 주는 값을 함수로 전달하는 역할을 한다.

JS가 하는 것은 bro의 값을 potato(인자)라는 것에 넣고, console에 있는 potato는 출력되는 것임

이것이 외부에 있는 데이터를 읽는 함수를 만드는 방법이다.

#2-1.1

console.log('hello!!',name," you have ",age,"years of age");

console.log('hello!!'+name+" you have "+age+"years of age");

여기에서는 +한 부분이 띄어쓰기가 제대로 안돼 있다.

JS에서는 “”와 ‘’모두 string이다.

``(백틱)

console.log(`hello!! ${name} you have ${age} years of age`);

#2-2

HTML이 자바스크립트하고 함께 쓰려고 하면 어떻게 동작하는지 볼 수 있다.

예시로, h1을 선택해서 바꿀 수 있는 거 같이 말이다.

Title을 id로 선택해 볼 수 있다.

CSS에서 title id를 선택하려면 #title{} 하면 된다.

JS에서 id로 선택하려면 어떻게 해야 할까?

const title= document.getElementById("title");

console.log(title);//<h1 id="title">This Work</h1>

1. 가져온 title의 무엇인가를 바꾸려고 해보자.

2. JS는 나의 Html태그를 가져다가 객체로 만들 것이다. [DOM(document object module)이라고 하는 것 같음]

여기 있는 모든 것(=document 에 있는 모든 것)은 다 객체가 될 것이다.

우리가 찾게 될 모든 객체들의 함수들 DOM형태로 변경 가능하다.

웹 페이지에서 JS로 선택한 것은 DOM 즉 객체가 된다.

#2-3

document.title="I love you"; document Html의 title을 JS로 변경 할 수 있다.

Js로 Html을 수정할 수 있다.( 예를들어,class를 추가 하고, 어떻게 숨기고 보여줄 수 있는지)

\*\*querySelector는 노드의 첫 번째 자식을 반환한다.

const title= document.querySelector(“.title”);

위 명령어의 결과는 title이 null이므로 tilte을 html에서 받아오지 못했다고 할 수 있음

Title이 값을 받아오려면 어떻게 해야 할까?

const title= document.querySelector(“#title”);(id로 값을 불러오려면 #을 써야한다.)

#2-4

Js는 단지 html과 css를 바꾸고 싶어서 만들어진 것이 아니다.

이벤트에 반응하기 위해서 만들어진 것 이다.

이벤트: 웹사이트에서 발생하는 것들을 의미함(resize, click..등)

좋은 사실은 우리가 이벤트를 가로챌 수 있다는 것

window.addEventListener("resize",handleResize);

우리가 이벤트 받기를 기다리는 것(listen to event)

우리는 분명 어떤 이벤트인지 말해야 한다(여기서는 resize)

Listener: 이벤트가 발생 되었을 때, 실행하는 함수를 의미 한다

handleResize()로 쓰는 것이 아니다. handleResize()는 지금 바로 함수를 호출 하라는 뜻

handleResize로 쓰는 이유는 함수를 내가 필요한 시점에 호출하기 위해서 쓰는 것이다.

function handleResize(event){// 여기의 event는 어디서 온 것 일 까?->JS에서 왔디.

Js는 이벤트를 다룰 함수를 만들 때 마다 자동적으로 함수에 객체를 붙인다.

console.log(event);

}

function handleClick(event){

title.style.color= "red";

}

title.addEventListener("click",handleClick);//title은 이벤트가 발생하는 것을 기다리고 있다.

한번 클릭 해서 색깔이 바뀌면, 색깔이 바뀐 상태로 머물러 있게 된다.

#2-5

If-else

If(조건){

Block//조건이 참이면 실행한다}

else{

block//조건이 거짓이면 실행한다.}

JS에서는 ==이===이다.

피연산자

True && true=true; True || true=true;

True && false=false; True || false=true;

false && true=false; false || true=true;

false && false=false; false || false=false;

prompt(“Ask your name”)=> alert의 일종으로, 엄청 오래된 JS임

#2-7

이 단계에서 원하는 것은< h1 id="title">This Works!</h1>을 <h1 id="title"class=”clicked”>This Works!</h1>로 바꾸는 것이다. 이를 위해 css에서 .clicked클래스를 선언한 것이다.

Color가 뭔지 css가 뭔지 등등 아무 것도 신경 쓰지 않도록 위의 과정을 하는 것이다.

<h1 id="title"class="btn">This Works!</h1>이렇게 바꾸고, css에 .btn을 추가해 주면 this work 에 마우스를 가져가면, 손가락 모양이 생긴다. 하지만 this work를 클릭하면 <h1 id="title"class="clicked">This Works!</h1>로 바뀌면서, 손가락 모양은 없어진다.

* 이를 클래스를 변경하게 되면 이전 클래스를 존중하지 않는 사례라고 한다.

classList는 메소드(함수)를 가지는 것이다.

#2-7-1

title.classList.remove(CLICKED\_CLASS);을 했는데 이전 상태로 돌아가지 않는다. 왜냐하면, btn class를 제거할 수 없기 때문이다. Clicked 가 아닌 add로 인해 btn clicked 가 되어 버렸기 때문이다.

위의 조건문에서 우리는 하나의 클래스만 체크할 수 있으므로, contains이라는 함수가 필요하다.

Contains는 value가 존재하는지 아닌지를 체크한다.

const hasclass=title.classList.contains(CLICKED\_CLASS);hasclass에 true/false를 저장한다.

#2-7-2

Toggle은 위의 #2-7-1을 요약한 것이다.

toggle안의 값을 체크해서 클래스가 없으며 add 있으면 remove.

#3-1

const clockContainer=document.querySelector(".js-clock"),

* querySelector는 element의 자식을 탐색한다.
* Js-clock이라는 class이름을 찾아 줄 것이다.

clcokTitle=clockContainer.querySelector("h1");

* 이 경우, 나는 document로부터 보고 싶지 않으면 querySelector는 element의 자식을 탐색한다. clockContainer의 자식(=js-clock)을 탐색하고 싶어한다.
* <h1 class=”js-title”></h1>…????

콘솔에서 const date =new Date();

Date.mintue()이 업데이트 되지 않는 이유, 여기의 Date는 선언 당시의 시각으로 굳어져 있기 때문에 분이 업데이트 되지 않는다.

#3-2

setInterval(함수,함수의 실행시간(밀리세컨드))

clockTitle.innerHTML=`${hours}:${minutes}:${seconds<10? `0${seconds}`:seconds}`;

만약 seconds가 10보다 작으면 왼쪽 출력 10보다 크면 오른쪽 출력

#3-3

Form을 만들기

쿼리셀렉터올은 모든 것을 가져온다. 클래스명에 따른 엘라먼트들을 가져오는데, 이건 array를 준다.

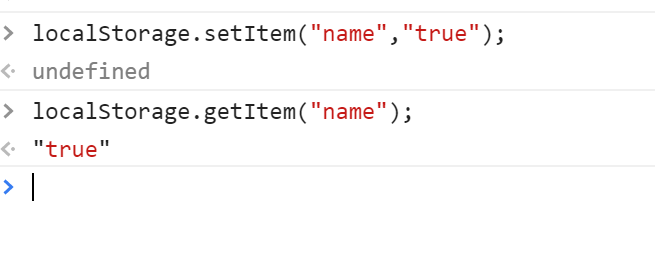
* 니콜썜은 쿼리 셀렉터 올을 쓰지 않는 다고 한다, 찾은 것이 유일하게 하나의 클래스명이라고 해도, array 안에 넣을 것이기 때문

쿼리셀렉터는 찾은 첫 번째 것을 가져온다. 클래스 CSS방식으로, 클래스 태그, 아이디 이런 식으로..

Localstorage:작은 정보를 나의 유저 컴퓨터에 저장하는 방법

F12누르고, 크롬으로가서 Application>localstorage

localStorage.setItem(“nico”,true);->로컬 스토리지 변수를 선언하는 것



클래스는 점(.), 아이디는 (#)잊지 말자!!!

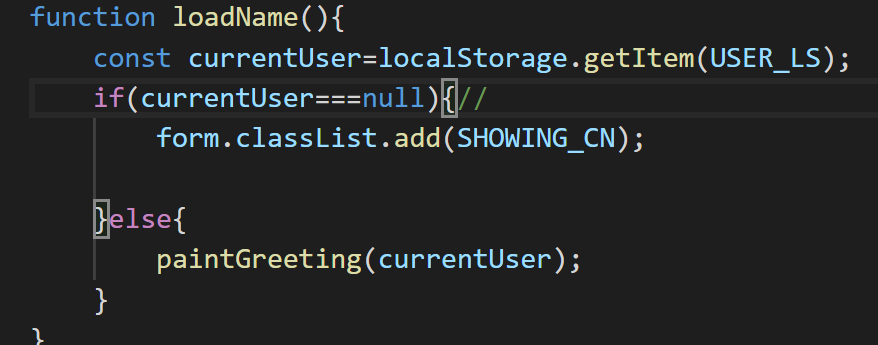
<form class="js-form form">

                   <input type="text" placeholder="what is your name?"/>

            </form>

            <h4 class="js-greetings greetings"></h4

여기서 form과 h4는 클래스이다. Input은 form의 자식으로 태그인 듯 하다.



왜 조건문의 null이 먹히지 않는지 고민해 보자

* currentUser가 남아 있기 때문이다. 크롬에서 currentUSer를 완전히 삭제해야 조건문이 먹힌다.

#3-4

이름을 요청하고 Enter를 누른다(Enter를 누른다는 것은 form을 제출하는 것을 의미 한다)

* default에의해, 너의 값을 다른 곳으로 보내려고 하기 때문에 새로고침 한 상태가 된다.

무엇인가를 form에 submit하면, handleSubmit이라는 function과 실행된다. 이 function은 전에 만들어둔 event와 같이 실행된다.

Event는 일종의 거품 같은 것이다.

* Event가 발생하면, root에서 일어나고, form에서 일어나게 된다.
* 이벤트가 올라가면서, 다른 모든 것들이 event에 반응하게 된다.
* Form을 제출하는 event가 발생하면, event가 계속 위로 올라간다 document까지
* 그 document는 다른 곳으로 갈 것이다.
* (Enter를 누르면, 프로그램 되어진 대로 이벤트는 다른 곳으로 가고, 페이지가 새로 고침 되는 것이다. =이것이 이벤트의 기본 동작이다.)

Event의 preventDefault는 이벤트의 기본 동작을 막아주는 함수이다.

저장은 localstorage에 하는 것!

#3-5

To do list만들기

만약, 작은 모듈을 만드는 것을 배우고 싶으면, 노마드 아카데미에 있는 Zero to here js과정을

시청하기 바란다!

const li=document.createElement("li");-> html에서 list를 생성하는 명령줄

html의 span은 div와 p태그와 함께 웹 페이지의 일부분에 스타일을 적용시키기 위해 사용된다. span태그안에 아무런 컨텐츠도 들어가지 않는다면, 해당 부분은 아무런 변화가 없지만, span내부에 객체가 들어가게 되면, CSS와 JS로 그 부분을 변형시키는 것이 가능해진다.

* 아래에서부턴, index.html의 #1을 만들어야 하는 과정을 담은 것이다.

li.appendChild(span);//span을 li의 자식 요소로 넣는다.

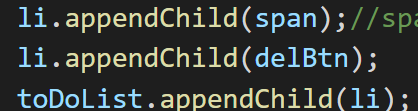
li.appendChild(li의 자식); ->li의 자식 노드 중 마지막 자식으로 붙이는 것

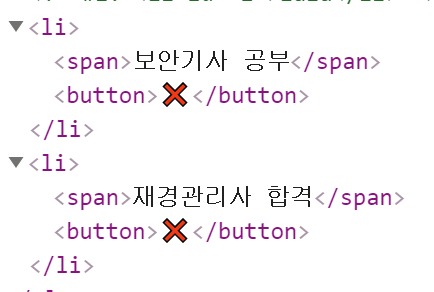
function paintToDo(todo) //#1은 비어있는 li와 버튼,span을 만들고 나서, span과버튼을 li에 넣는다.

금방 만든 li를 toDoList에 넣는다.

Js-toDoList >> li 이 순이다.

엔터를 눌렀을 때, li , delete, span 생성-> span과 delete버튼을 append-> 마지막으로 li를 ul(js-toDoList)에 넣는 것이다.

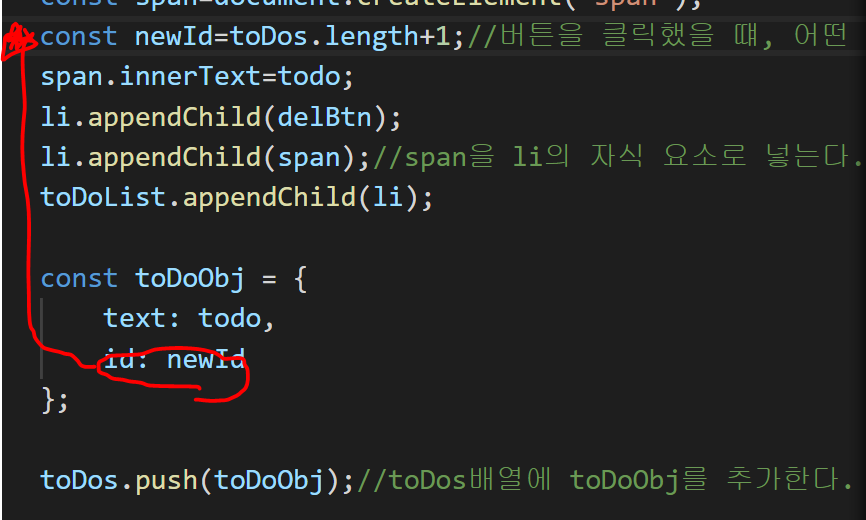




이거 위아래 왜 바꾸는지 모르겠다.

emty여야 하는데 empty가 아니라고 하면서 순서를 바꿔야 한다는데 무슨 소리인지 모르겠다.

#3-6

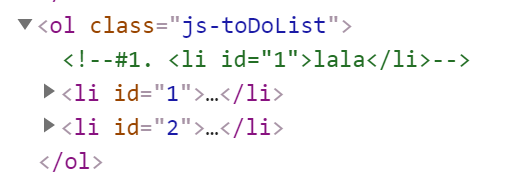


지금 li들에게는 id가 없기 때문에, 위의 작업을 해 주어야 한다.

* 삭제 버튼을 클릭했을 때, 어떤 li를 삭제 할지 알아야 하기 때문에
* Li의 태그를 ol로 바꾸면 안되는 것일까?(아마 HTML의 번호 보여주는 것이기에 번호는 매겨져 있지 않아서 안될 것 같다.) 아래와 같이 되므로, 하는 것 같다.



li.id=newId;를 하면, 아래와 같이 된다.

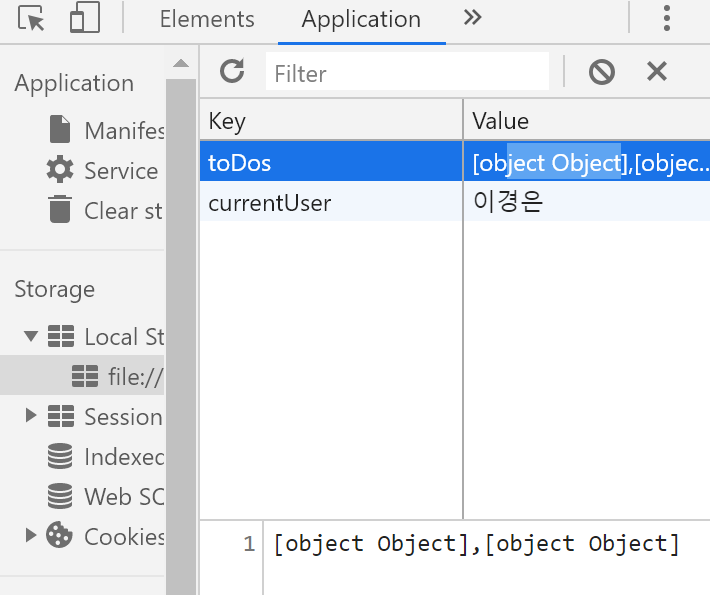


toDo를 왜 배열에 넣어서 저장하는 가?

* 그 이유는, local Storage에도 todo를 저장해둬야 하기 때문이다.

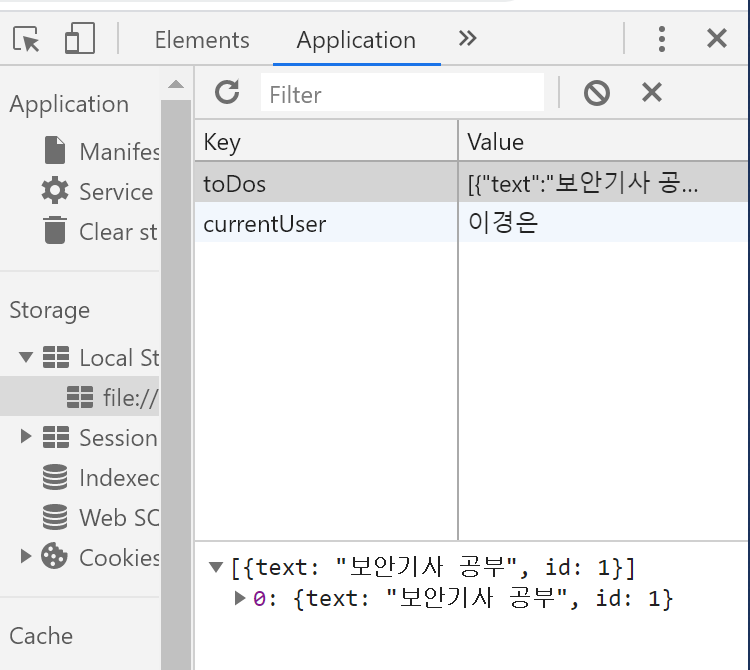
localStorage에서는 JS의 데이터를 저장할 수 없다.

* JS는 local storage에 있는 모든 데이터를 string으로 저장 하려고 한다.
* localStorage.setItem(“hello”,”true”);라고 저장을 해도, 여기서의 hello의 true는 boolean타입이 아닌, string이다.



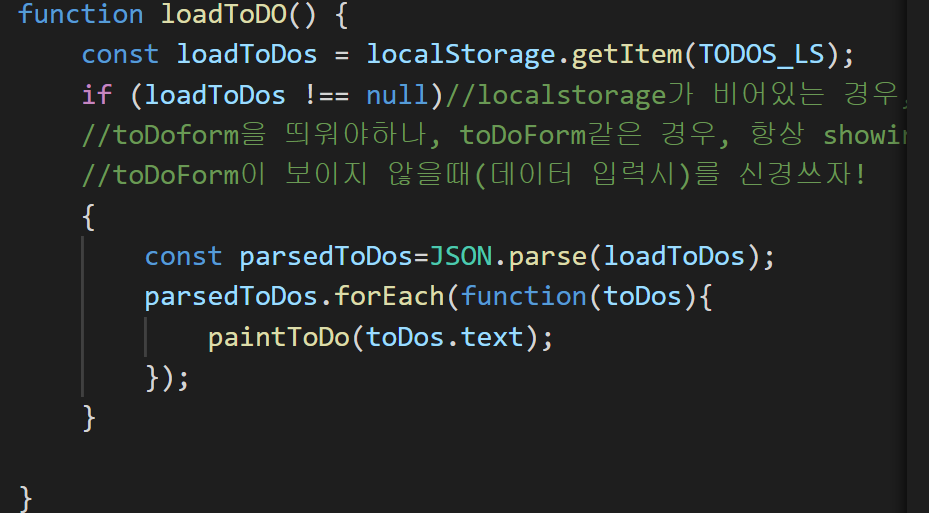
우리는 우리의 object가 string이 되도록 만들어야 한다.

JSON.stringify를 쓸 것 이다.



localStorage에서 JS로 불러온 toDos의 데이터는 object가 아닌 string이다. 우리가 js에서 object로 인식 할 수 있도록 하기 위해서는 아래의 명렁어를 작성해 준다

const parsedToDos=Json.parse(loadToDO);



forEach는 array를 위한 function이다. array각각의 요소들이 forEach()안의 함수를 호출한다.

Function something(toDos){  
 paintToDo(toDos.text);

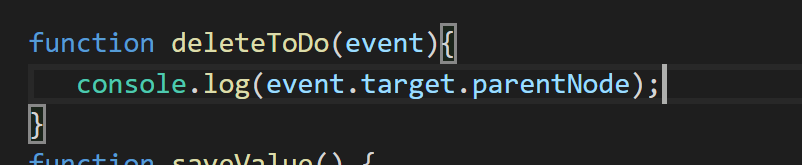
}

ParsedToDos.forEach(someting);

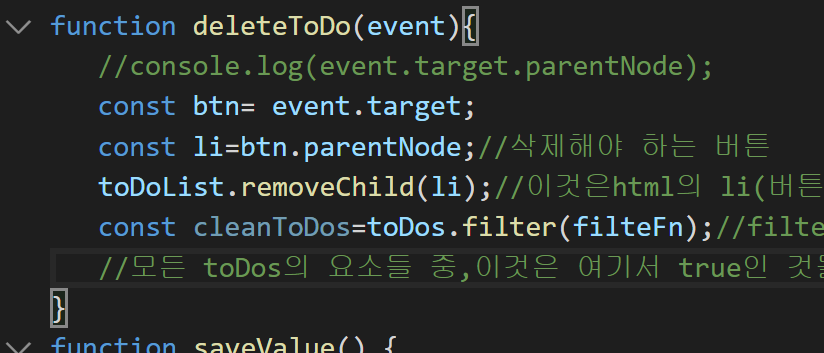
위 사진의 부분을 위의 text로 바꿔서 이해하자면,(사진과 텍스트의 형식만 다를 뿐, 기능은 동일하다.) pasedToDos(toDos)의 하나의 요소가 someting함수를 호출하는 것이다.

* toDos를 가져 온 뒤, 이 라인에서는 parse 즉, 가져온 것을 JS object로 변환해 줄 것이고,
* 각각에 대해서 paintToDo라는 function이 실행된다.
* Array도 function이 있고, string도 function이 있고, object도 function이 있다.

#3-7



Event.target.parentNode 이 명령어는 버튼을 클릭했을 때, 이 버튼의 부모를 찾아주는 역할을 한다. 이 버튼의 부모는 li이다.



Filter는 array의 모든 아이템을 통해 함수를 실행하고, 그리고 true인 아이템들만 가지고 새로운 array를 만든다.

* CleanTodo와 filter가 하는 것은 filterFn이 체크가 된 아이템들의 array를 주는 것이다.
* filter하고 forEach : 이것들이 list에 있는 모든 item을 위한 함수를 실행시키는 것이다.

#3-8

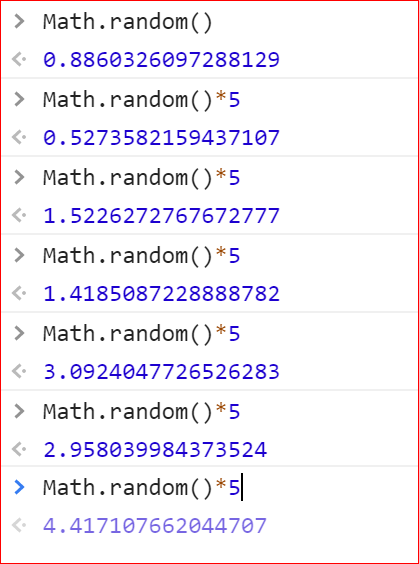
JS에서는 math라는 모듈이 있다.

이 모듈 안에는 많은 것 들이 있다.

그 중에서, 우리는 random이라는 함수를 공부 할 것이다.

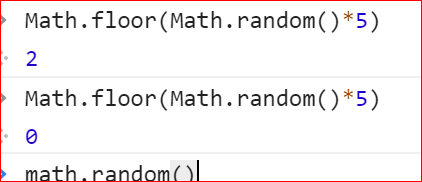
Random: 이 함수를 사용하면 임의적으로 숫자가 생성

내가 1-5 사이의 값을 출력 시키기를 원한다면, Math.random()\*5를 하면 된다



Math.floor(3.9): 소수점 자리를 모두 버림

Math.cell(3.9): 소수점 자리를 올림



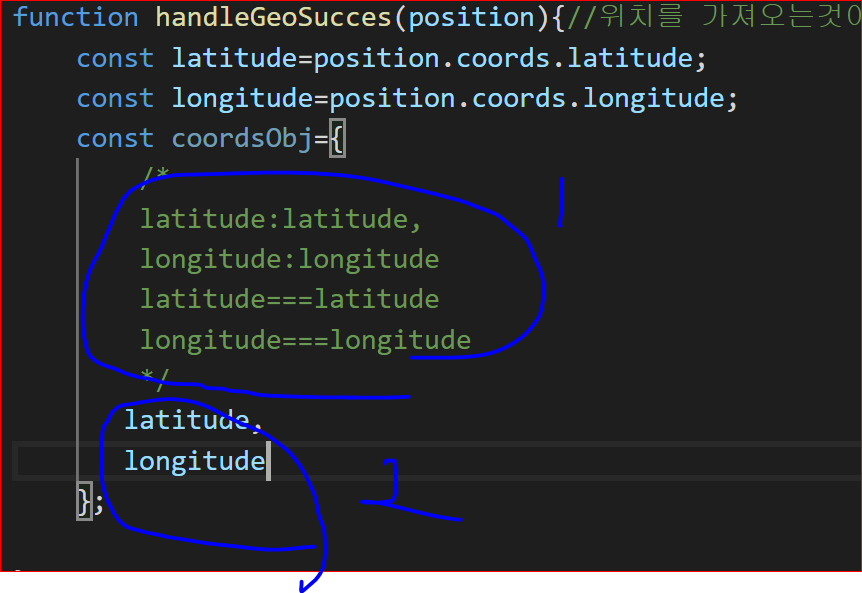
#3-9

Location정보를 가져오는 것: 유저의 location좌표 정보를 읽어서, 그 정보를 저장하고,

만약 이미 저장한 값이 없다면, 요청하고 저장 값이 없으면 하지 않아도 된다.

JS의 작은 팁: 객체에 변수의 이름과 객체의 key의 이름을 같게 저장할 때는 아래의 파란색 동그라미 두 개와 같이 지정한다

* 아래 객체의 용도는 좌표를 저장하기 위한 용도 이다.



이 강좌에서는 이거 말고 다른 data는 Api에서 받아올 일은 없다.

풀스택 유튜브 클론??<<-이게 뭘까?

여러 서버의 api에서, 더 많은 데이터를 가져오는 법을 배울 수 있다.

#3-10ㅇㄱㄷ

우리가 이용하는 웹사이트:<https://home.openweathermap.org/api_keys>

API(Application Programming Interface): 다른 서버로부터 손쉽게 데이터를 가져올 수 있는 수단

API를 제공하는 웹사이트를 이용하면, 해당 웹사이트를 통해 데이터를 얻을 수가 있는데,

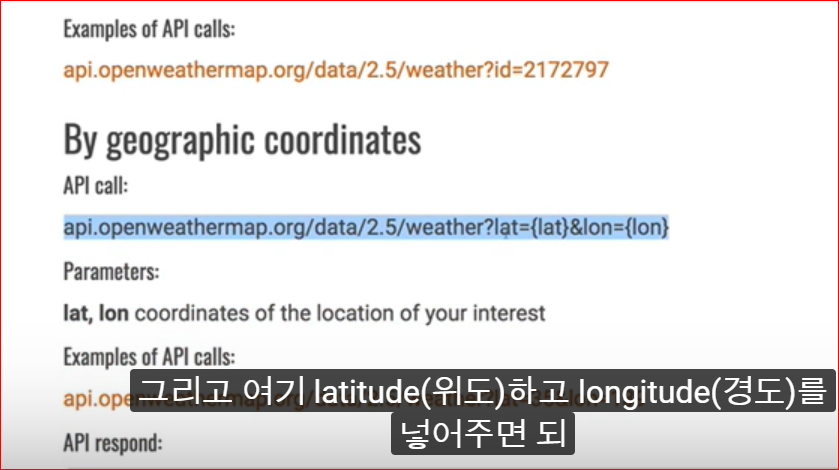
우리는 데이터만 가져올 것이다. 디자인이나 다른 것은 필요하지 않는다. 그 이유는

API는 특정 웹사이트로부터 데이터를 얻거나 컴퓨터끼리 소통하기 위해 고안된 것이다.

이 API를 개발한 사람들은 이 웹 페이지를 통해서 여기 데이터가 있고, 열려있으니 쓰고 싶은 사람 쓰라고 알려주는 것이다.

이 웹 페이지의 request를 누르면, 실제 데이터는 Application의 Todo들 저장한 것과 같은 데이터 형식이다.(그때 만든 것과 똑같은 데이터 형식이라고 보면 된다. 왜냐하면 이건 문자로 된 자바스크립트 오브젝트이다.)

* 우리가 이 웹사이트에서 날씨 정보를 찾아 볼 수도 있고, 이렇게 object로 된 날씨 데이터를 얻을 수도 있다는 것이다.
* 위의 행동이 가능하게 하기 위해서는 URL을 호출 해야 한다.

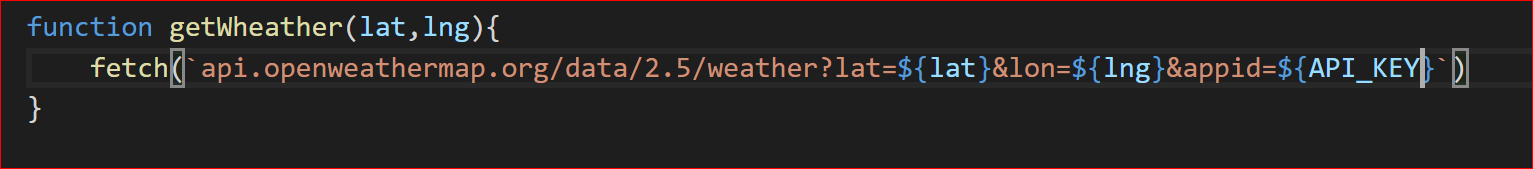


어떻게 JS를 이용해서 특정URL을 호출 할까? -> 사실 이 부분이 바로 JS가 강력해진 이유라고 할 수 있다.

왜냐하면 JS는 웹사이트로 Request를 보내고 응답을 통해서 데이터를 얻을 수 있는데, 가져온 데이터를 새로고침 없이도, 나의 웹사이트에 적용 시킬 수 있기 때문이다. ( ??? 사실 무슨말인지 …)

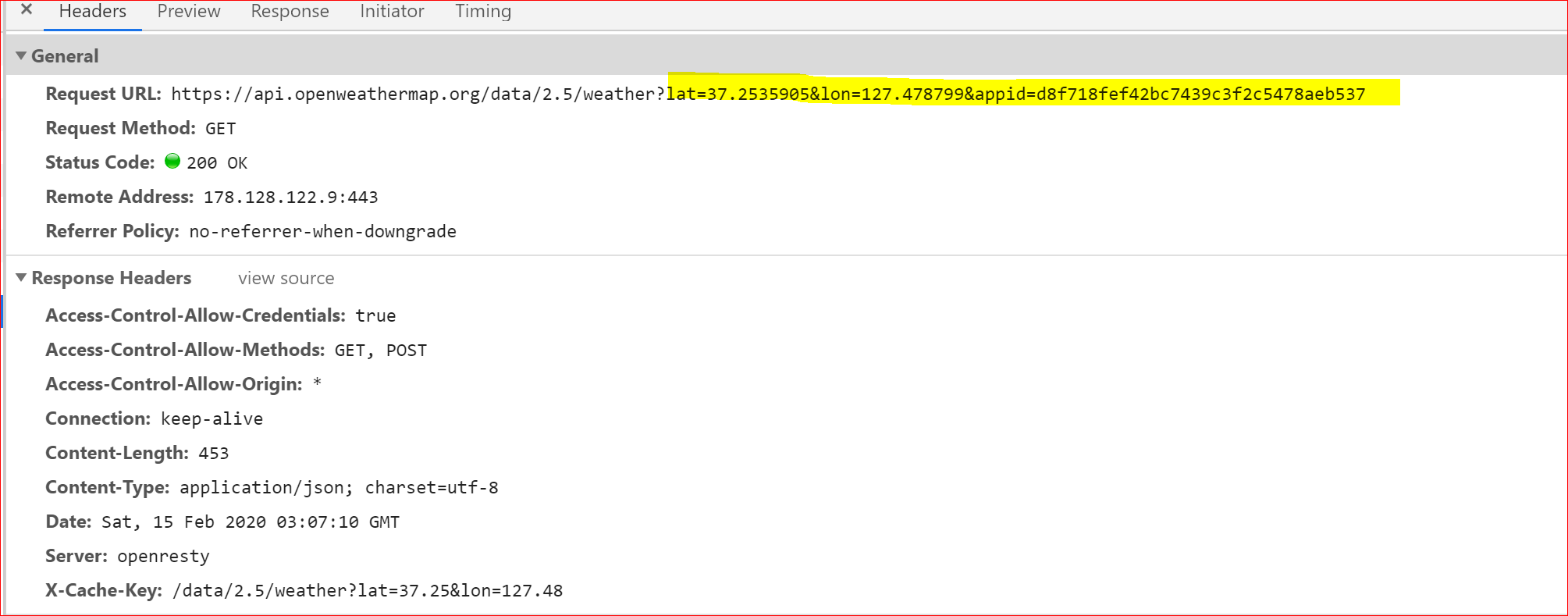
예를 들어, 우리가 이메일을 확인 할 때, 웹사이트를 새로고침 하지 않아도 실시간으로 메일이 오는 것을 확인할 수 있는 이유이다.

* JS가 보이지 않는 곳에서 계속 데이터를 가져오고 있기 때문이다.  
  그것을 지금 우리가 해야 할 것이다. 새로고침 없이 데이터를 가져올 것 이다.
* 그러기 위해서는 getWeather()에서,feth라는 함수를 써준다(데이터를 가져오기 위해서 사용하는 것임).
* Fetch()안에는 가져올 데이터가 들어가면 된다. 앞에 <https://넣어주고>, 따움표가 아닌 backtic(``)을 사용하자



이렇게 하면, API를 제공하는 쪽에서, 요청자의 API key를 통해서, 요청자가 무리한 요구를 하지는 않는지 알 수가 있다. 왜냐하면, 이것은 다 공짜이기 때문이다. 그러니까 자신들의 서버에 무리가 갈 만큼 사용하게 할 수는 없는 것 이다.

만약 local storage에 좌표 값이 없으면, askFforCoords함수가 실행되고, 이 함수 안에서 정상적으로 위치 정보를 가져오게 되면 handleGeoSuccess가 실행되는데 이 안에서 API가 최종적으로 호출 되기 때문이다.



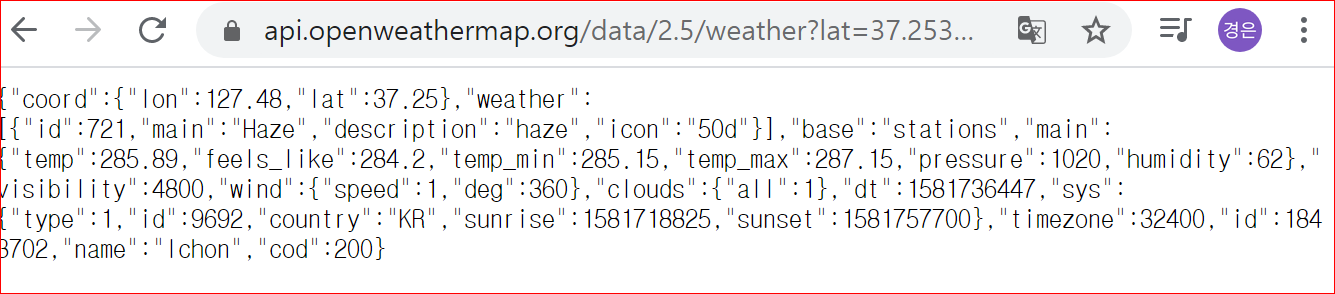
위의 사진을 보면, latitude 값과 longitude값도 있고, Apkey도 있다. 잘 동작하고 있는 증거이다.

위의 사진은 크롬의 개발자도구의 network패널이다 network패널은 우리가 request한 내용을 보여주는 것이고, 이것이 우리가 request한 내용이다.



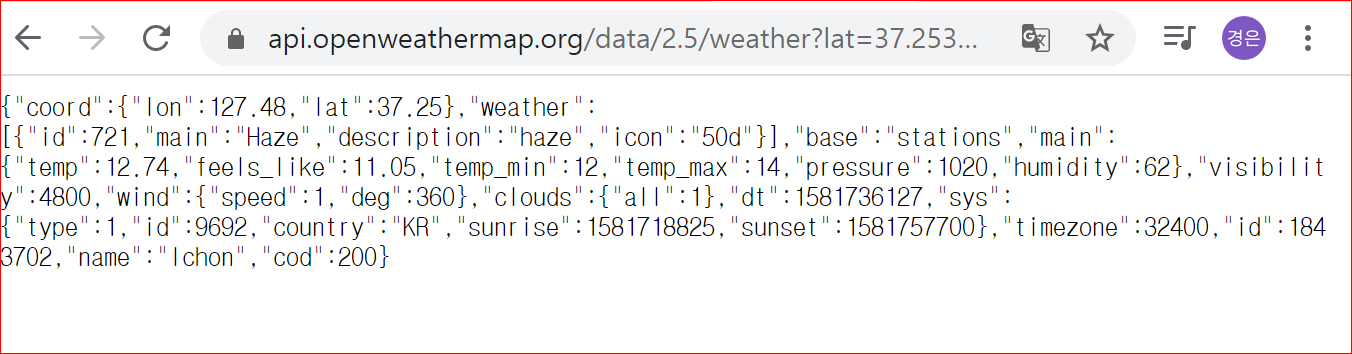
위의 사진은 거기에 대한 response도 보여준다.

URL이 가리키는 주소로가 보면 아래와 같은 정보를 주는 것을 알 수 있다.



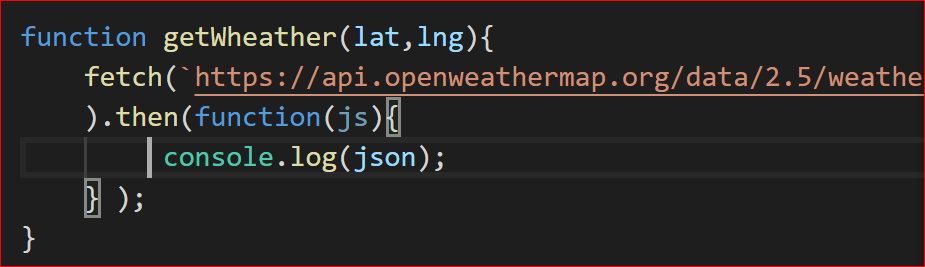
&uints=metric

을 추가 후,



Fetch(`------`).then() ->then은 함수를 하나 호출 할 것이다. then이라는 함수가 어떤 함수를 호출하는 데는 데이터가 우리한테 넘어 왔을 때 이다. 왜냐하면, 데이터가 들어오는데 시간이 걸리는 경우가 있기 때문이다.

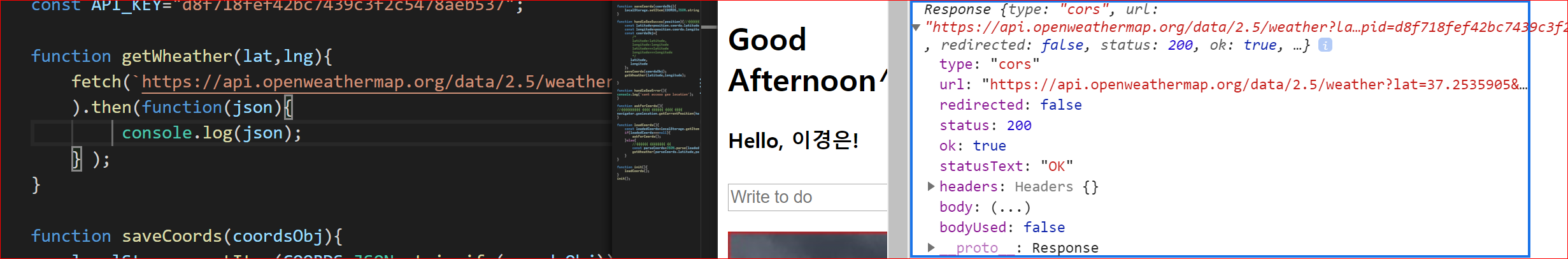
* Then() 의 역할: 데이터가 완전히 들어온 다음 함수를 호출해 주는 것



위의 함수에서 then을 사용하는 이유: fetch가 완료되기를 기다려야 하기 때문

만약, fetch를 기다리지 않고, 다음 작업을 지시하면, 다음 작업은 fetcch가 완료되길 기다리지 않을 거고, fetch는 정상적으로 완료되지 않을 수 있다.

* 그래서, 우리는 서버로부터 데이터가 들어올 때까지 기다려야 된다.



최종 완성본 이다. 온도와 위치에는 왜 CSS 적용이 안될까?

